

# Microcontrollers Fuzzy Logic and 16-bit MCUs: A Matter of Intuition

Michael Thompson

Philips Semiconductors-Microcontroller Group  
811 E. Arques Avenue, Sunnyvale, CA 94088-3409  
tel: 408-991-2000, fax 408-991-2828  
email: michael.thompson@sv.sc.philips.com

## Abstract:

*Most control applications involve the specification of a relationship between sensor signals and actuator outputs. Fuzzy logic provides an intuitive way to accomplish that. It allows the user to use linguistic rules to specify a nonlinear mapping between sensor signals and actuator outputs, thus provide a framework for programming an embedded system. Using a multi-joint robot system as a testbed, we implemented fuzzy logic on an 8051 compatible 16-bit microcontroller. The robot controlled by the MCU running the fuzzy logic algorithm is able to carry out a goal-directed motion sequence behavior. An 8-bit controller is also used to directly interface with the robot and communicate with the 16-bit MCU through I2C. In addition to carrying out AD/PWM conversions, the 8-bit also implements multiple loops of linear feedback for servo positioning and compliance control. This application note will demonstrate the implementation of fuzzy logic in an embedded control solution using a 16-bit microcontroller.*

## Introduction

Fuzzy logic was originally created as a mathematical model of human thought. It is said that fuzzy logic is able to capture the "vagueness" and "inexactness" of the concepts that we use for reasoning. In the past few decades or so, the main area of success with fuzzy logic has been in industrial control. The application of fuzzy logic allows us to specify the relationship between sensor inputs and actuator outputs using "if...then..." type of linguistic rules.

A fuzzy logic algorithm would be able to translate or interpolate these rules into a nonlinear mapping between sensor-input signals and actuator outputs for feedback control [1]. Fuzzy logic makes it easy for a human designer to fine-tune a control system through trial and error. Together with some other approaches such as artificial neural networks, genetic algorithm etc., fuzzy logic is considered a useful tool for non-model based control system design\*.

There are a number of software products available that would allow the user to design a fuzzy controller interactively with a special graphic user interface (GUI). These tools would usually generate C codes, which can be modified to fit into a user target platform. If you have to determine the parameters of your fuzzy logic control system on trial-and-error basis, it is certainly desirable to have some kind of graphic user interface so that you do not have to go into your code and make modification here and there.

As the number of inputs to a control system increases, the number of potential useful fuzzy rules increases dramatically and it

---

\* Non-model based design is a design that does not depend on a mathematical description of the plant dynamics.

becomes increasingly desirable to use some kind of automated method for rule synthesizing. There are a variety of such methods for doing this and active research is being carried out in this area currently. For example, the combination of fuzzy logic with artificial neural networks, genetic algorithm and learning automata have been proved to be effective in many applications.

In this paper, we will demonstrate the use of fuzzy logic in a 16-bit MCU. With a two-joint robot system as the testbed, we will discuss how to use fuzzy logic to tackle a specific control problem as well as some general programming issues related to the MCUs. Instead of exploring all the options that are out there, we will focus on one effective solution in this paper to get the readers quickly acquainted with the technique.

#### Robust control of a "bug" like robot leg

The two joint robot leg is powered by two gearmotors and it has a passive foot like structure at the end of the distal segment. We call the distal segment the "tibia", and the proximal segment the "femur" after animals. The behavioral purpose of this robot is to grab an object within the space it can reach. The location of the object is unknown to the robot and changing periodically. This is very similar to a situation when an insect walking over a very rough terrain is trying to find an object (such as a tree branch or twig ) to grab onto as a foothold. In this design, range sensing such as vision is not involved in the search of the object, as is the case with insects. Insects have developed a behavior where they use their legs as probes to actively sense where the object is and then establish a foothold onto it through a simple reflex [2]. The

active sensing reflex makes the "substrate-finding" behavior quite robust.

The robot is equipped with two potentiometers that give angular position readings for the two joints. On the two segments of the leg, strain gauges are pasted as force or touch sensors. The two strain gauges that are pasted near the junction of each actuator and the corresponding leg segment give us indications of the output torque of the two actuators. Three additional strain gauges are pasted along the distal segment (the tibia). These strain gauge readings can be decoded to determine where the touch between the leg and an external object has occurred. One of the strain gauges is pasted at the foot ankle region to signal foothold.

The purpose of controlling this leg is to replicate the "substrate-finding" behavior described above in a robust and reliable fashion. The challenge of this control problem lies in the fact that the position and touch sensors do not passively tell where the object is. The robot has to carry out active search movement to find out where the object is. In such a case, there is no way to linearly combine the sensor signals (or their derivatives and integrations) to produce the desired motor movement as in PID control. It is therefore an ideal application for us to try out fuzzy logic.

#### Outline of the approach

Two microcontrollers on an I2C bus are used to control this robot. Firstly, an 8-bit microcontroller is used to interface directly with the robot. In addition to carrying out the necessary A/D and PWM functions for sensor and actuator interface, this 8-bit microcontroller also implements position and force feedback to shape the actuator

dynamics so it becomes a position servo with proper compliance and damping properties. This is very important because the compliance in the actuators allows the robot to carry out contact based maneuvers reliably. Together with the sensors and actuators of the robot leg, the 8-bit MCU implements "virtual muscles" as seen from the microcontroller at the upper level, which is a 16-bit microcontroller running the fuzzy logic algorithm. we chose a Philips XA as the fuzzy logic engine because of its higher arithmetic capability. The 16-bit MCU reads "crisp" sensor values from the 8-bit microcontroller through I2C interface and converts them into fuzzy membership grades. These values are evaluated by a set of fuzzy rules implemented in the 16-bit MCU in order to generate appropriate motor commands that are sent back to the 8-bit MCU through I2C. With I2C, we can easily put multiple robot legs in the control system. For example, we can put together a six-legged hexapod robot.

In this application, the use of fuzzy logic and 16-bit MCUs is not intended to replace low level linear, classical control carried out with an 8-bit controller. Instead, we use fuzzy logic in an augmented and distributed fashion. Fuzzy logic in 16-bit controllers and linear classical control in 8-bit function in parallel and contribute to different aspects of the control.